Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of the Claims

Please cancel claims 14-15, 17-21, and 25-26 without prejudice.

Please amend claims 23, 24, 27 and 28 as follows:

1-13 (Cancelled)

14. (Cancelled)

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Cancelled)

19. (Cancelled)

20. (Cancelled)

21. (Cancelled)

22. (Cancelled)

23. (Currently Amended) ~~The method of claim 21~~, A method for playing back a recorded data stream from a storage medium, wherein the data stream has been recorded in data blocks, the method comprising the steps of:
   - reading payload blocks until a defect block is detected;
   - upon detection of the defect block, jumping back to a replacement block and

recovering the defect payload block by reading the replacement block;

- skipping the already read blocks; and
- continuing the reading of not yet read payload blocks;

wherein the replacement block is read and buffered and further payload blocks are read until the defect block is detected.

24. (Currently Amended) The method of claim 23 [[21]], wherein the read payload blocks are buffered and wherein a defect block is skipped and the following payload blocks and a parity block are read and buffered and wherein the defect payload block is reconstructed by using the buffered blocks and the parity block.

25. (Cancelled)

26. (Cancelled)

27. (Currently Amended) The method of claim 23 [[21]], wherein the blocks are clusters for a Blu-ray Rewritable Disc.

28. (Currently Amended) An apparatus equipped to perform the method of claim 23 [[21]], comprising:

- reading means for reading payload blocks and a replacement block for a defect payload block, wherein the replacement block is read and buffered and further payload blocks are read until the defect block is detected ;
- recovery means for recovering the defect block by using the read and buffered replacement block;
- skipping means for skipping the already read blocks; and
- control means for causing the reading of not yet read payload blocks to be continued.